Journal of Engineering and Technology for Industrial Applications

ITEGAM-JETIA

Manaus, v.12 n.57, p. 277-286. January/February, 2026.

DOI: https://doi.org/10.5935/jetia. v12i57.2892

ISSN ONLINE: 2447-0228

RESEARCH ARTICLE OPEN ACCESS

OPTIMIZATION OF CONVOLUTIONAL NEURAL NETWORKS THROUGH SOFT VOTING ENSEMBLE AND FINE-TUNED APPROACH TO IMAGE CLASSIFICATION

Johan Wijaya Kusuma*[1], Julian Supardi[2]

[1, 2]Department of Informatics Engineering, Sriwijaya University, P.O BOX 30129, Palembang, Indonesia.

[1]https://orcid.org/0009-0004-7906-4097, [2]https://orcid.org/0000-0002-5836-9236

Email: *kusumajohanwijaya@gmail.com, julian@unsri.ac.id

ARTICLE INFO

Article History
Received: November 5, 2025
Revised: December 10, 2025
Accepted: January 1, 2026
Published: January 31, 2026

Keywords:
ResNet,
EfficientNet,
DenseNet,
Ensemble Learning,
Image Classification.

ABSTRACT

This study introduces an optimization method to enhance image classification accuracy using a Soft Voting Ensemble of fine-tuned convolutional neural networks (CNNs). The goal is to evaluate whether combining multiple CNN architectures can outperform individual models. Three pre-trained networks DenseNet, ResNet, and EfficientNet were fine-tuned through transfer learning on a Kaggle image dataset. To improve generalization, data augmentation techniques such as random rotation, flipping, and zooming were applied. After fine-tuning, the models were integrated using a soft voting strategy that averages prediction probabilities to determine the final class. Performance was tested on 94 unseen images, fully separated from the training data. The individual accuracies were 91.5% for DenseNet, 87.2% for ResNet, and 93.6% for EfficientNet. The proposed ensemble achieved the highest accuracy of 94.7%, with precision, recall, and F1-score all reaching 0.95. These findings indicate that the ensemble approach successfully combines the strengths of different CNNs, reduces classification errors, and increases model robustness. Overall, the soft voting ensemble provides a reliable, scalable, and effective solution for improving CNN-based image classification, especially when dealing with limited or diverse datasets.

I. INTRODUCTION

In recent years, the fields of computer vision and deep learning have experienced significant advancement, driven by the rapid growth of computational power and the availability of large-scale annotated datasets. Deep learning models, particularly convolutional neural networks (CNNs), have become the backbone of modern image classification systems due to their ability to automatically learn hierarchical feature representations from raw pixel data. Unlike traditional image processing methods that depend on handcrafted features, CNNs can effectively capture spatial dependencies, textures, and semantic structures directly from the data. This capability has enabled breakthroughs in diverse domains such as medical imaging, autonomous driving, surveillance, agriculture, and industrial automation, where visual understanding plays a crucial role in decision-making processes [1].

The fundamental strength of CNNs lies in their hierarchical feature extraction capability, where low-level filters capture edges and textures, while deeper layers represent complex and abstract concepts. Over the past decade, numerous CNN architectures have been proposed to improve model depth, efficiency, and generalization. Among the most influential architectures are ResNet, DenseNet, and EfficientNet, which have become benchmarks in computer vision research and applications. ResNet introduced residual connections to address the vanishing gradient problem and enable the training of very deep networks [2]. DenseNet, on the other hand, employs dense connections among layers to promote feature reuse and gradient flow efficiency, leading to compact yet powerful models [3].

EfficientNet introduced a compound scaling technique that systematically balances network depth, width, and input resolution to achieve state-of-the-art accuracy with fewer parameters [4]. Despite their impressive performance, single CNN architectures still face several challenges. In practice, the performance of a CNN often depends on the quality, size, and diversity of the training dataset. When data is limited, noisy, or exhibits high intra-class variability, models tend to overfit or become biased toward dominant features. Moreover, a single model might excel at identifying certain visual patterns but perform poorly on others, depending on the nature of its learned representations. This model-specific bias often limits generalization, especially when deployed in real-world scenarios with unseen data distributions. Therefore, improving classification stability and robustness remains an open challenge in deep learning–based vision systems [5]. One promising direction to overcome these limitations is ensemble learning. Ensemble methods combine the predictions of multiple models to produce a more reliable and accurate output than any individual model could achieve alone. In traditional machine learning, ensemble strategies such as bagging, boosting, and stacking have demonstrated consistent improvements in performance and stability. In deep learning, similar principles can be applied by combining the outputs of multiple neural networks trained with different initializations, architectures, or subsets of data. Ensemble learning leverages the diversity among models both in terms of structure and learned representations to reduce variance and increase predictive robustness [6].

Among various ensemble strategies, soft voting has emerged as a particularly simple yet effective approach. In soft voting, the probabilistic predictions (output probabilities) from several base classifiers are averaged or weighted to determine the final class prediction. Unlike hard voting, which only considers the majority vote of discrete class labels, soft voting retains the confidence level of each model's prediction, allowing the ensemble to make more nuanced decisions. This approach is especially beneficial when combining heterogeneous deep models, as their probability distributions can complement one another, leading to improved accuracy and smoother decision boundaries [6]. In this study, a Soft Voting Ensemble approach is proposed to enhance image classification performance by combining multiple fine-tuned CNN models. Specifically, three high-performing architectures DenseNet, ResNet, and EfficientNet are fine-tuned through transfer learning on a custom image dataset obtained from Kaggle. Transfer learning allows leveraging pre-trained weights from large-scale datasets such as ImageNet, accelerating convergence and improving generalization even with limited target data. To further increase model robustness, data augmentation techniques are applied during training. Augmentation operations such as random rotation, flipping, scaling, and zooming help to artificially expand the training data and expose the models to diverse visual variations.

This not only mitigates overfitting but also improves the adaptability of CNNs to unseen real-world conditions [7]. The motivation behind using multiple CNNs in an ensemble lies in their complementary feature extraction capabilities. Each architecture has unique design principles that influence how it interprets image information. For instance, ResNet's skip connections enable efficient learning of residual mappings, making it particularly effective for recognizing global patterns and coarse structures. DenseNet's densely connected layers enhance information flow and encourage feature reuse, which benefits texture-rich or detailed images. Meanwhile, EfficientNet's compound scaling provides a balanced architecture that captures both fine and coarse details efficiently. By combining these models using a soft voting mechanism, the ensemble benefits from the diverse representation power of each CNN, leading to improved classification stability and accuracy across different image types. In practical applications, ensemble methods also provide an additional advantage: model uncertainty reduction.

Since deep networks are often sensitive to initialization and data variations, a single model might produce inconsistent predictions when exposed to slightly altered inputs. The ensemble approach mitigates this by averaging multiple predictions, which statistically reduces variance and enhances decision confidence. This makes soft voting particularly suitable for safety-critical applications such as medical diagnosis or industrial inspection, where prediction reliability is paramount [8]. This research aims to achieve three main objectives. First, to investigate the effectiveness of the soft voting ensemble in improving image classification accuracy compared to individual fine-tuned CNN models. Second, to evaluate the impact of transfer learning and data augmentation on model performance and generalization. Third, to analyze the complementary behavior among DenseNet, ResNet, and EfficientNet when combined under a unified probabilistic voting scheme. Through these objectives, this study contributes to the understanding of how ensemble deep learning can be leveraged for more robust and scalable visual recognition systems, particularly in domains where labeled data are scarce or heterogeneous.

II. MATERIALS AND METHODS

II.1 DATASET

The dataset utilized in this study comprises a single image classification dataset focusing on *Pempek*, a traditional Indonesian delicacy originating from Palembang. Figure 1 illustrates representative examples from the *Pempek* image classification dataset. The dataset was obtained from the publicly available Kaggle repository and consists of five distinct classes representing various types of Pempek: Pempek Lenjer, Pempek Kapal Selam, Pempek Adaan, Pempek Keriting, and Pempek Kulit. The images within the dataset exhibit considerable diversity in terms of lighting conditions, backgrounds, and camera orientations, thereby ensuring a comprehensive representation of real-world variations. Such heterogeneity is essential for enhancing the generalization capability of the proposed model.

Figure 1: Example Images.
Source: Authors, (2026).

To strengthen the dataset and mitigate the risk of model overfitting, several data augmentation techniques were employed, including random rotation, horizontal and vertical flipping, cropping, brightness adjustment, and image resizing. These transformations not only increased the effective dataset size but also improved the robustness of the model when exposed to unseen data. Prior to model training, a series of data preprocessing steps were conducted to standardize and optimize the dataset. These include:

- Resizing with Black Padding: All images were resized to a uniform resolution of 224 × 224 pixels. To preserve the original aspect ratio and prevent geometric distortion, any empty regions resulting from resizing were filled with black pixels.
- Data Augmentation: Augmentation operations were systematically applied to the training data to increase variability and improve the model's resilience to environmental and perceptual changes.
- Normalization: Each pixel value was normalized to a range of [0, 1] by dividing by 255. This normalization process ensures numerical stability and accelerates model convergence during training.

Initially, the dataset contained 485 original images. These images were randomly divided into 391 training samples (approximately 80%) and 94 testing samples (approximately 20%). To further improve the robustness of the model and mitigate overfitting, data augmentation was applied exclusively to the training set. Augmentation techniques included random rotation, horizontal and vertical flipping, cropping, brightness adjustment, and resizing. As a result, the number of training images increased from 391 to 2,904 augmented samples, thereby expanding the dataset's variability and improving the model's resilience to real-world conditions. The final dataset comprised a total of 2,904 images across five categories, as summarized in Table 1.

Table 1: Dataset.

Dataset	Source	Classes	Original Images	Test Images	Train Images	After Augmentation
Pempek Food Picture	Kaggle	5	485	94	391	2904

Source: Authors, (2026).

II.2 PROPOSED METHODS

II.2.1 CNN RESNET

ResNet (Residual Network) introduces the principle of residual learning to effectively mitigate the vanishing gradient problem that typically arises in deep convolutional neural networks [9]. By integrating shortcut or skip connections that bypass one or more layers, ResNet facilitates the training of very deep models while maintaining stability and accuracy. In this study, the ResNet50 architecture is employed, consisting of 50 layers organized into a series of convolutional and identity blocks that incorporate batch normalization and ReLU activation functions. Each residual block employs identity mappings and 1×1 convolutions to adjust feature dimensions, enabling efficient hierarchical feature extraction. ResNet50 offers a favorable balance between model complexity and classification accuracy, outperforming shallower variants such as ResNet34 in capturing high-level semantic features while remaining computationally practical for fine-tuning tasks [10]. The network is initialized using pre-trained ImageNet weights, which accelerates convergence and enhances the model's generalization performance within the transfer learning framework.

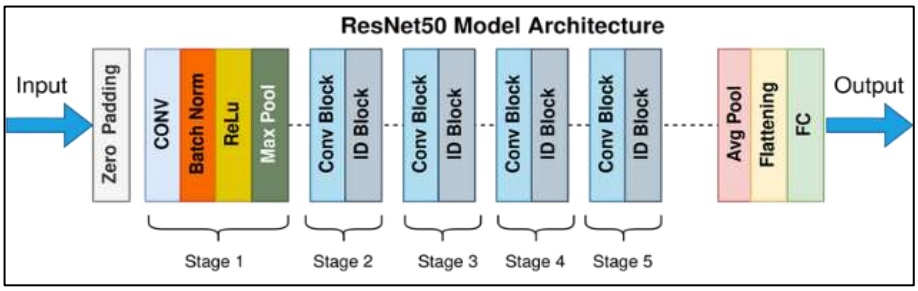

Figure 2: ResNet50 Architecture.
Source: Authors, (2026).

II.2.2 CNN Densenet

DenseNet (Densely Connected Convolutional Network) introduces a fully connected feed-forward mechanism within each dense block, where every layer is directly connected to all subsequent layers [11]. This design encourages feature reuse and efficient information flow, addressing the vanishing gradient issue and improving feature propagation throughout the network. The architecture utilized in this research is DenseNet121, which comprises 121 layers organized into four dense blocks, interleaved with transition layers responsible for convolution and pooling operations. Each dense block applies batch normalization, ReLU activation, and both 1×1 and 3×3 convolutional filters. This densely connected structure captures hierarchical feature representations efficiently, using fewer parameters compared to conventional CNNs. DenseNet121 was selected due to its high accuracy-to-parameter ratio and efficient memory utilization. Its compact architecture makes it particularly suitable for medium-scale datasets, providing robust classification performance and generalization capability [12].

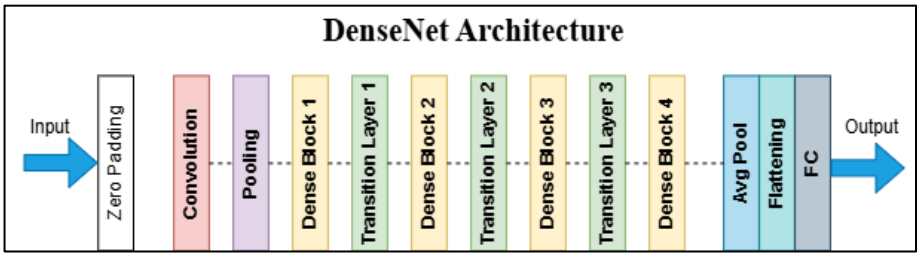

Figure 3: DenseNet121 Architecture.
Source: Authors, (2026).

II.2.3 CNN Effecientnet

EfficientNet introduces a compound scaling method that systematically and uniformly scales a network's depth, width, and input resolution using fixed scaling coefficients [13]. This balanced scaling strategy allows the model to achieve superior accuracy while maintaining computational efficiency. In this work, the EfficientNet-B0 variant is adopted as it represents the baseline model within the EfficientNet family. It utilizes MBConv (Mobile Inverted Bottleneck Convolution) layers combined with squeeze-and-excitation optimization to enhance representational power at a low computational cost. EfficientNet-B0 was chosen for its lightweight design and competitive performance, making it ideal for experimentation in environments with limited computational resources [14]. Despite having significantly fewer parameters than deeper CNNs, it delivers efficient inference and serves as a strong baseline for ensemble-based approaches.

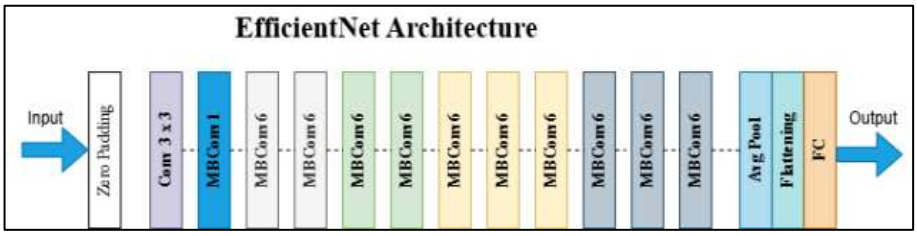

Figure 4: EffecientNet Architecture.
Source: Authors, (2026).

II.2.4 Fully Connected Layer

To ensure architectural consistency and improve feature discrimination across all CNN backbones (ResNet50, DenseNet121, and EfficientNet-B0), a uniform fully connected (FC) layer configuration was appended to the output of each network. The original top classification layers were removed, and the extracted feature maps from each pre-trained base model were passed through a Global Average Pooling (GAP) layer, followed by a series of dense layers with ReLU activation and dropout regularization to mitigate overfitting.

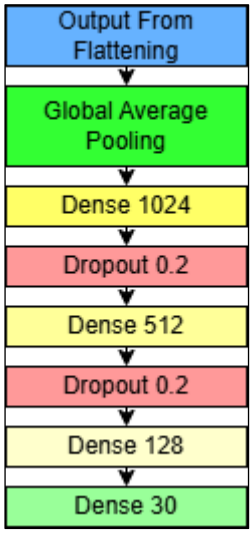

Figure 5: Fully Connected Layer.
Source: Authors, (2026).

This configuration maintains a balance between model expressiveness and computational efficiency. The GAP layer effectively reduces spatial dimensions while preserving essential semantic information, and the successive dense layers progressively abstract higher-level features [15]. Dropout regularization with a rate of 0.2 is applied to prevent neuron co-adaptation and enhance generalization [16]. By applying a standardized FC layer design across all CNN architectures, any performance variations can be attributed primarily to the representational capabilities of the backbone networks rather than inconsistencies in the classifier head. This methodological consistency aligns with best practices in transfer learning and ensemble modeling, ensuring fair and reliable model comparison [17].

II.2.5 Soft Voting Ensemble

Ensemble learning has emerged as an effective strategy to enhance the accuracy, robustness, and generalization capabilities of deep learning models. Among various ensemble approaches, soft voting combines the probabilistic outputs (class confidence scores) of multiple classifiers instead of relying solely on discrete class labels as in hard voting. This probabilistic aggregation enables the final prediction to reflect the relative confidence of each model across all possible classes, thereby producing a more balanced and informed decision. Previous studies have demonstrated that soft voting generally surpasses individual classifiers and hard voting ensembles by reducing model bias and variance, as well as by exploiting the complementary strengths of diverse architectures [18]. In this study, a soft voting ensemble is employed to integrate the outputs of three distinct CNN backbones—ResNet50, DenseNet121, and EfficientNet-B0—each possessing unique representational advantages. ResNet50 facilitates effective training of deep networks through residual learning; DenseNet121 enhances feature reuse and gradient flow via dense inter-layer connectivity; and EfficientNet-B0 provides a lightweight yet high-performing framework through compound scaling. By aggregating their probabilistic outputs, the ensemble model effectively leverages the individual strengths of each architecture while compensating for their respective weaknesses. This synergistic combination leads to improved classification accuracy, enhanced model stability, and better generalization performance across diverse data distributions.

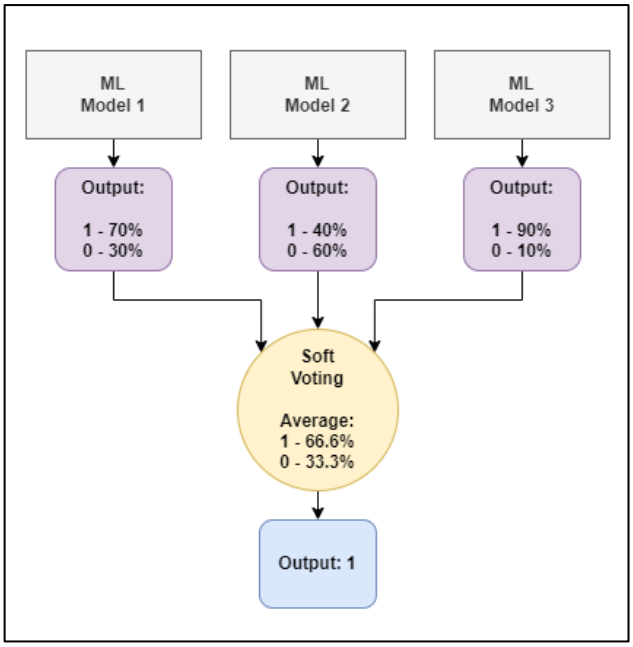

Figure 6: Fully Connected Layer.
Source: Authors, (2025).

III. RESULTS AND DISCUSSIONS

III.1 TRAINING MODEL

The training process for the CNN model used several techniques to enhance convergence and prevent overfitting. One key technique was employing an Exponential Decay learning rate schedule [19], starting with an initial learning rate of 0.001, which decreases by a factor of 0.96 every 1000 steps, allowing the learning rate to reduce gradually over the epochs. The Adam optimizer [20] was applied, using categorical cross-entropy as the loss function, suitable for multi-class classification tasks. To further improve generalization, early stopping [21] was implemented with a patience setting of 10 epochs, restoring the model weights with the best validation loss. Additionally, a model checkpoint callback was enabled to save automatically the model that achieved the highest validation accuracy during training. A detailed summary of these configurations is presented in Table 2.

Table 2: CNN Training Parameter.

Parameter	Value/Setting
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Learning Rate Schedule	Exponential Decay (initial = 0.0001, decay rate = 0.96, decay steps = 30)
Epochs (max)	100
Early Stopping	Patience = 10 (monitor = val_loss, restore_best_weights = True)
Model Checkpoint	Save best model (based on val_accuracy)
Batch Size	32

Source: Authors, (2026).

After each CNN model is independently trained and assessed, their predictions are integrated using a soft voting approach. For every input image, each model generates a probability distribution across the 30 distinct food categories. The soft voting mechanism then computes the average of these probabilities and ultimately selects the class with the highest average probability as the ensemble's final prediction. This method harnesses the collective "confidence" of the three models, resulting in a more dependable and precise final classification. The performance of this ultimate ensemble model is subsequently evaluated and benchmarked against the individual models using consistent metrics: Accuracy, Precision, and and Recall. Following the previously outlined training procedures, each fine-tuned CNN model (ResNet, EfficientNet, and DenseNet) underwent evaluation based on its training and validation performance. Figure 7 (ResNet), Figure 8 (EfficientNet), and Figure 9 (DenseNet) graphically represent the progression of accuracy and loss over epochs for each model, underscoring variations in their convergence rates and propensities for overfitting.

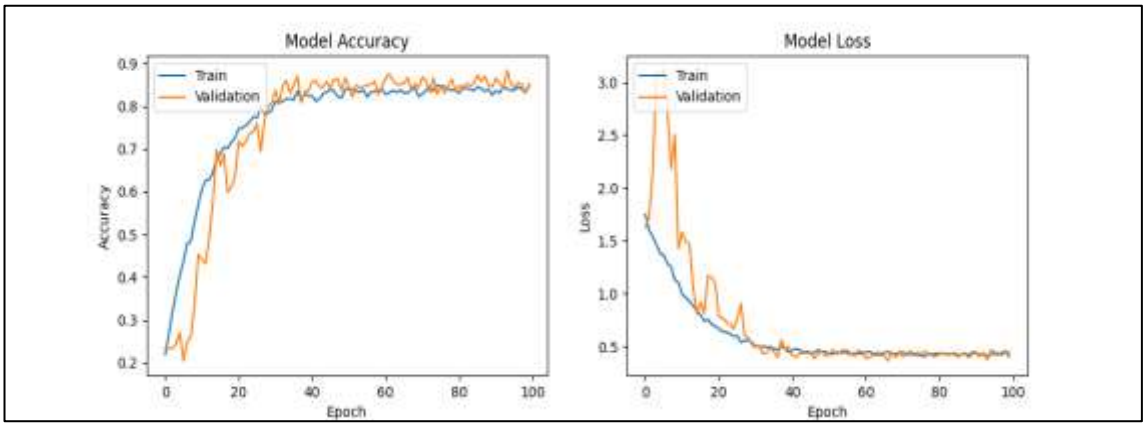

Figure 7: Accuracy and Loss Graph For ResNet.
Source: Authors, (2026).

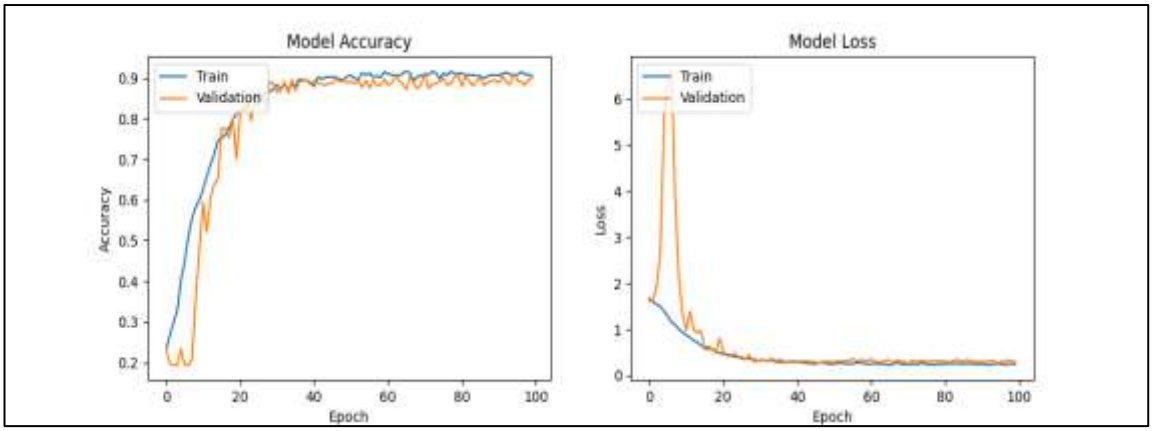

Figure 8: Accuracy and Loss Graph For EffecientNet.
Source: Authors, (2026).

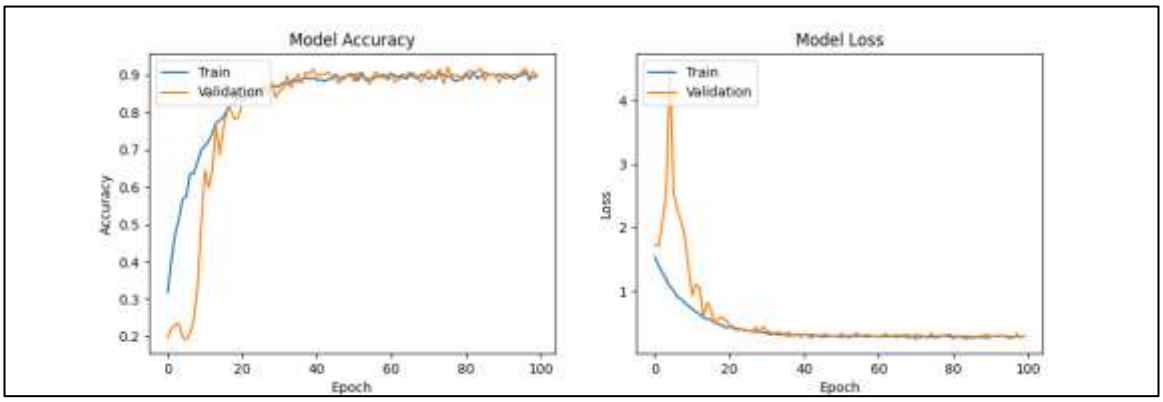

Figure 9: Accuracy and Loss Graph For DenseNet.
Source: Authors, (2026).

The performance graphs for ResNet, EfficientNet, and DenseNet models over 100 training epochs demonstrate successful and effective training for all three architectures. ResNet shows steady learning with training and validation accuracy stabilizing around 80-85%, and loss decreasing to approximately 0.4-0.5, indicating good generalization without significant overfitting. EfficientNet performs slightly better, with accuracy stabilizing between 85-90% and loss reducing to low values around 0.25-0.35, reflecting rapid convergence and excellent generalization. DenseNet also achieves competitive results, matching EfficientNet's accuracy range of 85-90%, with an initially more volatile but ultimately stable decreasing loss around 0.2-0.3, indicating robust and stable training. Across all models, the gap between training and validation metrics remains small, highlighting minimal overfitting and strong generalization capabilities. Overall, these results confirm that the fine-tuning strategies applied were highly effective, yielding accurate and reliable models suitable for the dataset.In summary, the graphs show that the three model architectures, ResNet, EfficientNet, and DenseNet, all trained very successfully. Each model demonstrated rapid convergence and did not show significant signs of overfitting. This confirms that the fine-tuning approach used is effective and well-suited for this dataset.

III.2 TESTING MODEL

After completing the training phase, each fine-tuned CNN model ResNet, EfficientNet, and DenseNet was tested on a dataset of 94 food images spanning 5 categories. This evaluation aimed to measure each model's ability to generalize to new, unseen data. The confusion matrices shown in Figures 10, 11, and 12 illustrate the results for each CNN architecture, providing a detailed view of correctly and incorrectly classified samples for each class. These matrices reveal the classification consistency and the types of errors produced by each model, forming the basis for comparing the performance of the three architectures.

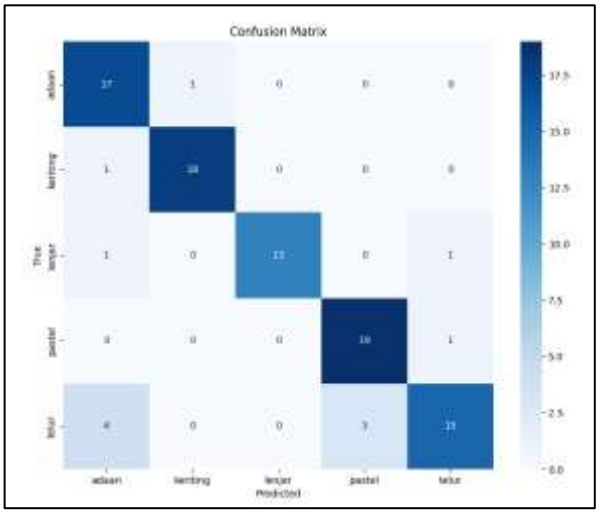

Figure 10: Confusion Matrix ResNet.
Source: Authors, (2026).

The confusion matrix for the ResNet model, presented in Figure 10, illustrates its classification performance across five distinct categories: 'adaan,' 'keriting,' 'lenjer,' 'pastel,' and 'telur.' Analysis of the matrix reveals robust performance for the 'keriting' and 'pastel' classes, with 18 and 19 instances correctly classified, respectively. Minor misclassifications for these categories include one 'keriting' instance erroneously predicted as 'adaan' and one 'pastel' instance misclassified as 'telur.' For the 'adaan' class, 17 instances were accurately identified, although one was incorrectly assigned to 'keriting.' Similarly, 'lenjer' exhibited 13 correct classifications, with individual instances being mislabeled as 'adaan' and 'telur.' Conversely, the 'telur' category posed the most significant challenge for the ResNet model. While 15 'telur' instances were correctly identified, a notable number of misclassifications occurred, with 4 'telur' instances incorrectly predicted as 'adaan' and 3 as 'pastel.' These observations suggest a potential ambiguity or feature overlap between 'telur' and the 'adaan' and 'pastel' categories, indicating areas where model refinement or a more diverse training dataset could enhance discriminative capabilities for these specific classes.

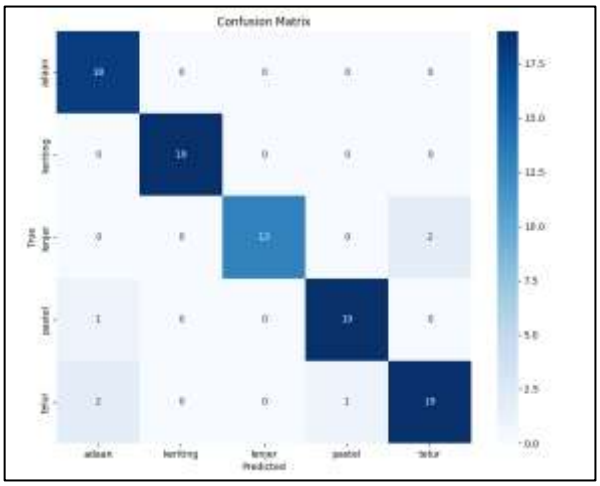

Figure 11: Confusion Matrix EffecientNet.
Source: Authors, (2026).

The confusion matrix for the EfficientNet model, depicted in Figure 11, provides insights into its classification efficacy across the 'adaan,' 'keriting,' 'lenjer,' 'pastel,' and 'telur' categories. The model demonstrates superior performance in classifying 'adaan' and 'keriting,' achieving 18 and 19 correct predictions, respectively, with no misclassifications observed for these categories. The 'pastel' class also exhibited strong classification, with 19 instances accurately identified, though one instance was incorrectly predicted as 'adaan.' For the 'lenjer' class, 13 instances were correctly classified, but two instances were erroneously assigned to 'telur.' The 'telur' category achieved 19 correct classifications; however, two instances were misclassified as 'adaan' and one as 'pastel.' These results collectively suggest that EfficientNet generally exhibits high accuracy across most categories, with 'lenjer' showing a slight tendency for misclassification into 'telur,' and 'telur' occasionally being confused with 'adaan' and 'pastel.' This detailed breakdown highlights EfficientNet's robust performance, while also pinpointing specific areas where further optimization might yield marginal improvements in discriminatory power.

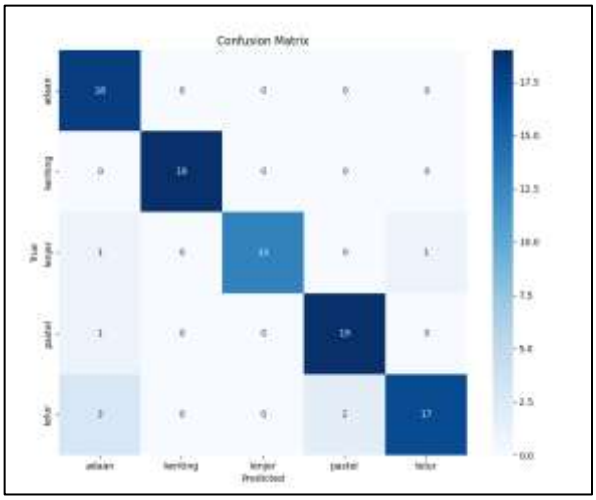

Figure 12: Confusion Matrix DenseNet.
Source: Authors, (2026).

The confusion matrix for the DenseNet model, illustrated in Figure 12, delineates its classification performance across five categories: 'adaan,' 'keriting,' 'lenjer,' 'pastel,' and 'telur.' DenseNet demonstrates strong classification for 'keriting' and 'pastel,' achieving 19 correct predictions for each with no misclassifications. For the 'adaan' class, 18 instances were correctly identified, with one instance incorrectly predicted as 'lenjer' and one as 'pastel.' The 'lenjer' class shows 13 correct classifications, but one instance was misclassified as 'adaan.' The 'telur' category correctly classified 17 instances, yet three instances were erroneously predicted as 'adaan' and two as 'pastel.' Overall, DenseNet exhibits robust performance, particularly for 'keriting' and 'pastel.' However, similar to the other models, 'adaan,' 'lenjer,' and 'telur' occasionally experience misclassifications. Overall, the fine-tuned ResNet, EfficientNet, and DenseNet models demonstrated commendable classification performance across the five distinct food categories (adaan, keriting, lenjer, pastel, and telur).

This is evident from the consistently high values observed along the diagonal elements of their respective confusion matrices (Figures 10, 11, and 12), which signify accurately classified instances. Specifically, EfficientNet exhibited exceptional precision for 'adaan' and 'keriting,' achieving zero misclassifications in those categories, while both ResNet and DenseNet showed strong performance, particularly for 'keriting' and 'pastel.' These outcomes collectively underscore the models' successful capability to discern and generalize the unique visual characteristics of each food type following the fine-tuning process. Nevertheless, a minor proportion of misclassifications persisted in certain categories across all three models. For instance, ResNet occasionally confused 'telur' with 'adaan' and 'pastel,' while EfficientNet showed a tendency to misclassify 'lenjer' as 'telur,' and 'telur' as 'adaan' and 'pastel.' DenseNet, although robust, still exhibited some instances where 'adaan,' 'lenjer,' and 'telur' were misclassified.

While each individual Convolutional Neural Network (CNN) architecture namely ResNet, EfficientNet, and DenseNet achieved high classification accuracy, their respective confusion matrices consistently revealed a minor but persistent number of misclassifications across various food categories. These subtle inconsistencies underscore that each model possesses distinct strengths and weaknesses in feature representation, highlighting the potential for complementary performance gains. To address these minor classification errors and further enhance the overall predictive performance, a Soft Voting Ensemble approach was strategically implemented. This robust method operates by combining the probabilistic outputs generated by the three best-performing models: ResNet, EfficientNet, and DenseNet. By averaging their individual prediction probabilities before a final class determination is made, this ensemble strategy effectively leverages the complementary capabilities inherent in each network.

This synergy significantly boosts the model's robustness and improves its generalization capacity when presented with unseen test data. The performance of the Soft Voting Ensemble model was rigorously evaluated using a comprehensive suite of classification metrics: accuracy, precision, recall, and F1-score. The results, meticulously detailed in Table 3, demonstrate the model's high effectiveness. The ensemble achieved an impressive accuracy of 0.95 (95%), indicating that 95% of all instances were correctly classified. Furthermore, the model exhibited a precision of 0.95 (95%), signifying that among all instances predicted as a specific class, 95% were indeed correct. A recall of 0.95 (95%) highlights the model's ability to correctly identify 95% of all actual instances within each class. The F1-score, also at 0.95 (95%), further corroborates the balanced and high performance of the ensemble, reflecting a strong harmony between precision and recall. These metrics were derived from a test set comprising 94 total samples.

Table 3: Soft Voting Model Performance.

Metric	Value
Accuracy	0.95 (95%)
Precision	0.95 (95%)
Recall	0.95 (95%)
F1-Score	0.95 (95%)
Total Samples	94

Source: Authors, (2026).

These results are consistent with the confusion matrix shown in Figure 13.

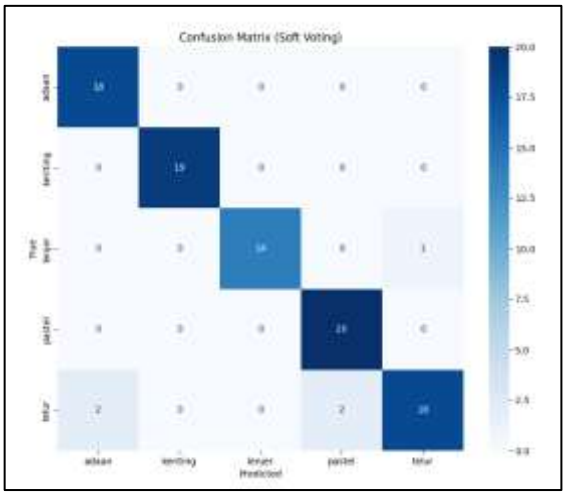

Figure 13: Confusion Matrix SoftVoting.
Source: Authors, (2026).

The matrix illustrates a highly effective classification with the majority of instances correctly predicted for each class. Notably, the 'pastel' category demonstrates perfect classification, with all 20 actual 'pastel' instances correctly identified. Similarly, the 'keriting' category exhibits strong performance, accurately classifying all 19 instances. The 'adaan' category shows only 2 misclassifications, with two 'adaan' instances being incorrectly predicted as 'telur'. For the 'lenjer' category, 14 instances are correctly identified, but 1 instance is misclassified as 'telur'. In the 'telur' category, 18 instances are correctly classified, while 2 are misclassified as 'adaan' and 'pastel' respectively. Despite these minor misclassifications in the 'adaan', 'lenjer', and 'telur' categories, the overall reduction in errors compared to the individual models' confusion matrices is substantial. This outcome unequivocally underscores the efficacy of the Soft Voting Ensemble approach in leveraging the complementary strengths of ResNet, EfficientNet, and DenseNet models. The observed performance metrics and the detailed confusion matrix analysis collectively validate the ensemble's superior robustness and generalization ability on unseen test data, making it a highly reliable solution for the task of accurate food image classification.

IV. CONCLUSIONS

In summary, this study successfully developed a highly accurate and robust image detection and classification system. The experimental results demonstrate that fine-tuning modern CNN architectures yields strong baseline performance; however, incorporating an ensemble learning strategy—specifically through soft voting—further enhances model accuracy and generalization. By integrating multiple architectures with complementary feature extraction capabilities, the ensemble effectively mitigates individual model limitations and achieves superior predictive performance.These findings underscore the significant potential of ensemble-based approaches in addressing complex classification challenges within the domain of computer vision.

For future work, it is recommended to evaluate the proposed ensemble framework using larger and more heterogeneous datasets, as well as under more challenging environmental conditions such as variations in illumination, background complexity, and image noise. Such investigations would provide deeper insight into the scalability, robustness, and real-world applicability of the soft voting ensemble method.

V. AUTHOR'S CONTRIBUTION

Conceptualization: Johan Wijaya Kusuma, Julian Supardi.
Methodology: Johan Wijaya Kusuma, Julian Supardi.
Investigation: Johan Wijaya Kusuma.
Discussion of results: Johan Wijaya Kusuma, Julian Supardi.
Writing – Original Draft: Johan Wijaya Kusuma.
Writing – Review and Editing: Johan Wijaya Kusuma, Julian Supardi.
Resources: Johan Wijaya Kusuma.
Supervision: Julian Supardi.
Approval of the final text: Johan Wijaya Kusuma, Julian Supardi.

VI. ACKNOWLEDGMENTS

This research was independently funded by the researcher without any external financial support.

VII. REFERENCES

[1] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," Sep. 01, 2017, MIT Press Journals. doi: 10.1162/NECO_a_00990.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, [Online]. Available: http://arxiv.org/abs/1512.03385

[3] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," Jan. 2018, [Online]. Available: http://arxiv.org/abs/1608.06993

[4] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," Sep. 2020, [Online]. Available: http://arxiv.org/abs/1905.11946

[5] C. Zhang, O. Vinyals, R. Munos, and S. Bengio, "A Study on Overfitting in Deep Reinforcement Learning," Apr. 2018, [Online]. Available: http://arxiv.org/abs/1804.06893

[6] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," Aug. 2022, doi: 10.1016/j.engappai.2022.105151.

[7] H. B. Kibria, M. Nahiduzzaman, M. O. F. Goni, M. Ahsan, and J. Haider, "An Ensemble Approach for the Prediction of Diabetes Mellitus Using a Soft Voting Classifier with an Explainable AI," Sensors, vol. 22, no. 19, Oct. 2022, doi: 10.3390/s22197268.

[8] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," Nov. 2017, [Online]. Available: http://arxiv.org/abs/1612.01474

[9] C. L. Lin and K. C. Wu, "Development of revised ResNet-50 for diabetic retinopathy detection," BMC Bioinformatics, vol. 24, no. 1, Dec. 2023, doi: 10.1186/s12859-023-05293-1.

[10] F. Putra Panghurian, H. Pranoto, E. Irwansyah, and F. S. Pramudya, "Comparison of Resnet Models in UNet Classifier for Mapping Oil Palm Plantation Area with Semantic Segmentation Approach," 2024. [Online]. Available: www.ijacsa.thesai.org

[11] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," Aug. 2016, [Online]. Available: http://arxiv.org/abs/1608.06993

[12] Y. Hou, Z. Wu, X. Cai, and T. Zhu, "The application of improved densenet algorithm in accurate image recognition," Sci Rep, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-58421-z.

[13] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," May 2019, [Online]. Available: http://arxiv.org/abs/1905.11946

[14] I. P. A. T. Wahyudi, I. G. I. Sudipa, L. G. B. Libraeni, M. L. Radhitya, and I. M. D. P. Asana, "Performance Comparison of MobileNet and EfficientNet Architectures in Automatic Classification of Bacterial Colonies," Indonesian Journal of Data and Science, vol. 6, no. 2, pp. 333–342, Jul. 2025, doi: 10.56705/ijodas.v6i2.218.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018.

[16] I. Salehin and D. K. Kang, "A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain," Jul. 01, 2023, Multidisciplinary Digital Publishing Institute (MDPI). doi: 10.3390/electronics12143106.

[17] F. Zhuang et al., "A Comprehensive Survey on Transfer Learning," Nov. 2019, [Online]. Available: http://arxiv.org/abs/1911.02685

[18] M. A. Khan, N. Iqbal, Imran, H. Jamil, and D. H. Kim, "An optimized ensemble prediction model using AutoML based on soft voting classifier for network intrusion detection," Journal of Network and Computer Applications, vol. 212, Mar. 2023, doi: 10.1016/j.jnca.2022.103560.

[19] Z. Li and S. Arora, "An Exponential Learning Rate Schedule for Deep Learning," Nov. 2019, [Online]. Available: http://arxiv.org/abs/1910.07454

[20] S. Dereich and A. Jentzen, "Convergence rates for the Adam optimizer," Jul. 2024, [Online]. Available: http://arxiv.org/abs/2407.21078

[21] M. K. Anam, S. Defit, Haviluddin, L. Efrizoni, and M. B. Firdaus, "Early Stopping on CNN-LSTM Development to Improve Classification Performance," Journal of Applied Data Sciences, vol. 5, no. 3, pp. 1175–1188, Sep. 2024, doi: 10.47738/jads.v5i3.312.